A structured and systematic model-based development method for automotive systems, considering the OEM/supplier interface.

Kristian Beckers[a], Isabelle Côté[b], Thomas Frese[c], Denis Hatebur[b,d], Maritta Heisel[d]

[a]*Technische Universität München, Germany*
[b]*Institut für technische Systeme GmbH, Germany*
[c]*Ford Werke GmbH, Germany*
[d]*paluno - The Ruhr Institute for Software Technology University Duisburg-Essen, Germany*

Abstract

The released ISO 26262 standard for automotive systems requires to create a hazard analysis and risk assessment and to create safety goals, to break down these safety goals into functional safety requirements in the functional safety concept, to specify technical safety requirements in the safety requirements specification, and to perform several validation and verification activities. The experience shows that the definition of the technical safety requirements and the planning and execution of the validation and verification activities has to be done jointly by the OEMs and the suppliers. In this paper, we present a structured and model-based safety development approach for automotive systems. The different steps are based on Jackson's requirement engineering. The elements are represented by a UML notation extended with stereotypes. The UML model enables a rigorous validation of several constraints. We illustrate our method using a three-wheeled-tilting control system.

Keywords: ISO 26262, automotive, hazard analysis, risk assessment, safety goal, safety, functional, technical, requirement, UML, validation and verification

1. Introduction

Developing and constructing road vehicles has become a complex task due to the increase of features, such as adaptive cruise control or lane keeping assist functions. The safety aspects of these features have to be taken into account during the product development. Another fact is that most of these complex systems are distributed. Distributing the system amongst the different parties involved means that the overall system is broken down into several components and/or subsystems. Different divisions

Email addresses: `beckersk@in.tum.de` (Kristian Beckers), `i.cote@itesys.de` (Isabelle Côté), `tfrese@ford.com` (Thomas Frese), `d.hatebur@itesys.de` (Denis Hatebur), `maritta.heisel@uni-due.de` (Maritta Heisel)

within the OEM are responsible for the components / subsystems, which are provided by different suppliers.

This raises the complexity for the manufacturer (OEM), who has to organize the necessary activities. With the release of ISO 26262 - Road vehicles Functional safety in November 2011 [1], the automotive sector benefited from a consistent functional safety process for developing and constructing electric/electronic (E/E) systems. ISO 26262 addresses all levels of development, including definition of functions/features, systems engineering as well as details of software and hardware development. The standard should be applicable to different scenarios for establishing this process, including e.g., the OEM and any number of suppliers for the distributed systems.

Since ISO 26262 is a risk-based functional safety standard addressing malfunctions, its process starts with a hazard analysis to determine the necessary risk reduction to achieve an acceptable level of risk. The hazard analysis results in safety goals with an automotive safety integrity level (ASIL) that describes the necessary risk reduction. Performing such a hazard analysis is a challenging task because

- It should be comprehensible for different stakeholders, e.g., engineers, project leaders, managers.
- It should be possible to review the hazard analysis within a realistic time period.
- Hazard analyses of different projects should be comparable.
- In a hazard analysis, all relevant faults or situations need to be considered.

This hazard analysis is usually performed by the OEM division responsible for the development of the overall system.

According to ISO 26262, the next steps are to break down these safety goals into functional safety requirements. It has to be justified that the derived functional safety requirements are suitable to achieve the stated safety goals. These functional safety requirements are then detailed and the technical safety requirements are derived. In addition, the Verification and Validation (V&V) is performed. The results of the V&V activities is fed back and collected in an appropriate way to support the creation of the safety case.

Most of these complex systems are distributed. This distribution includes several challenges: For the requirement engineering, it has to be determined who has to provide which content at which level of detail. Usually, the OEM division responsible for the development of the system creates the logical architecture and then distributes requirements to different divisions within the OEM responsible for the components. These divisions receive all requirements from systems in which their component is involved in, integrate the requirements and cascade the requirements to the component suppliers. They do the implementation and supply pieces of hardware and software that then have to be integrated into the vehicle. Some of the requirements engineering (RE) has to be done by the OEM and the supplementary RE has to be added by the suppliers.

For the verification and validation (V&V), the OEM division responsible for the overall system has to ensure that the V&V tasks are defined and cascaded to the other divisions and the suppliers. Some aspects can only be validated on vehicle level by the OEM division responsible for the system (e.g. the overall behavior of the system), some aspects can be validated on component level by the divisions responsible for the components (e.g. the behavior of the component) and other aspects can only be validated using internal interfaces of the component by the suppliers. When the V&V is performed, the results of the V&V activities at suppliers side and within the different OEM divisions needs to be fed back and collected by the division responsible for the overall system.

In addition, heterogeneous and concurrent engineering processes, methods and tools exist within the affected parties which need to be harmonized. Communication between OEM and divisions/suppliers has to be organized via requirements as well as verification and validation documents.

In this paper, we propose a structured method based on UML models supported by a tool for the hazard analysis, the requirement engineering, and the V&V activities.

The advantage of a UML model-based approach is that the different artifacts are explicitly connected instead of having loosely coupled documents. On this overall model, consistency checks can be performed. These consistency checks can be specified with the Object Constraint Language (OCL) from the Object Management Group (OMG) [2].

Our paper is organized as follows: In Sect. 2, we introduce some background knowledge as well as previous work to establish a common understanding. Section 2.1 briefly introduces the underlying standard used throughout our method followed by a short description of the requirements analysis method in Sect. 2.2. The Framework, in which the method is embedded, is outlined in Sect. 2.3 and the model is introduced in Sect. 2.4.

Section 3 introduces the case study we use to illustrate our method. Section 3.1 describes the hazard analysis and risk assessment artifacts [1]. In section 3.2, the artifacts created in the functional safety concept is given [2]. The parts of the method that have already been published will only be briefly discussed. The interested reader can find more details in the provided citations.

In Section 4, the technical safety requirement specification method illustrated with the example is presented.

Section 5 introduces the applied support tool and Sect. 6 discuss related work. Finally, in Sect. 7, we provide a conclusion and an outlook on future work.

2. Background

2.1. ISO 26262

In 2011, the functional safety standard, ISO 26262 [3], was published. It is derived from the generic functional safety standard IEC 61508 [4] and aligns with the automotive safety life-cycle including specification, design, implementation, integration, verification, validation, configuration, production, operation, service, decommissioning, and safety management. ISO 26262 provides an automotive-specific risk-based approach for determining risk classes that describe the necessary risk reduction for achieving an acceptable residual risk, called *automotive safety integrity level (ASIL)*. The possible ASILs are *QM*, *ASIL A*, *ASIL B*, *ASIL C*, and *ASIL D*. The ASIL requiring the highest risk reduction is called ASIL D. In case of a QM rating, the normal quality

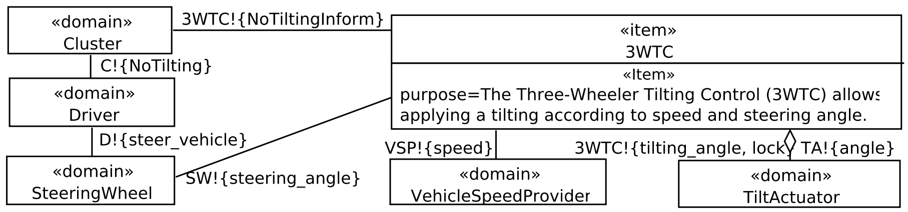

Figure 1: Context Diagram for 3WTC

measures applied in the automotive industry are sufficient. The standard also addresses the OEM-supplier interface to some extend. ISO 26262 Part 8 requires an appropriate definition (e.g. by using a development interface agreement) of the interface between OEM and supplier, but as the application of the standard should be possible in different project scenarios, the standard does not provide a predefined and dedicated method to split technical responsibilities amongst the different participating parties.

2.2. Requirements Analysis

Our requirements engineering method is inspired by and based on the approach proposed by Jackson [5]. In this approach, requirements can only be guaranteed for a certain context. Therefore, it is important to describe the *environment* in which the system to be build (called *item* in the automotive domain) will operate. This is achieved by a *context diagram*. Figure 1) shows an example of such a diagram. The context diagram consists of boxes representing different elements, also called *domains* (e.g. SteeringWheel in Fig. 1[1]), in the application environment that already exist.

A special domain is the system to be build, i.e., the item. The different domains are connected by interfaces consisting of shared phenomena. Shared phenomena may be events, operation calls, messages, and the like. They are observable by at least two domains, but controlled by only one domain. The phenomenon *steering_angle* is an example for such a shared phenomenon. It is observable by the domains *3WTC* (3-Wheeler-Tilt-Control system) and *SteeringWheel*. However, only SteeringWheel (SW) controls that phenomenon. This is indicated by the exclamation mark after the abbreviated name of the domain (see 'SW!{steering_angle}' in Fig. 1).

2.3. Functional Safety Framework

The Ford Integrated process for Functional Safety (FIFS) consists of templates, examples and guidelines in Microsoft Word and Microsoft Excel. These templates, examples and guidelines were developed and improved (using project feedback) since 2009. They were applied in more than 20 projects and cover all parts of ISO 26262 being relevant for an OEM who does not develop software and hardware. If the templates are applied according to the guidelines, ISO 26262 compliant (work) products are developed. The method is based on practical experience in the automotive domain.

[1] As a simplification, we assume that the domain SteeringWheel consists of the actual physical steering wheel as well as a steering wheel provider module.

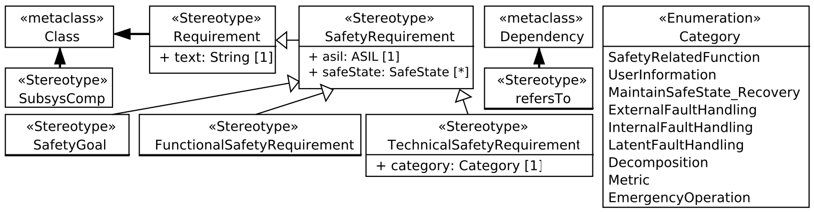

Figure 2: Profile Part concerning Requirements and Components

Within the V-model applied in ISO 26262, the first step of requirements engineering is to perform a hazard analysis and risk assessment for the system under consideration. Output of this step is given by the safety goals, describing the highest level of safety requirements. In the functional safety concept (FSC), the safety goals from the hazard analysis are broken down into functional safety requirements. These functional safety requirements are mapped to subsystems or components.

The task of the subsequent step is to split the functional safety requirements up into technical safety requirements. Within our approach, the technical safety requirement categories *SafetyRelatedFunction, UserInformation, MaintainSafeState_Recovery, ExternalFaultHandling, LatentFaultHandling, Decomposition*, and *Metric* are used. With these functional safety requirements and technical safety requirements, the requirement activities of the OEM are finalized within the setup chosen for our method. The technical safety requirements are cascaded to the other OEM divisions and finally to the suppliers as described in Sect. **??**. and the V&V phase is started.

The method presented in this paper supports the planning and performing of V&V activities as well as the documentation of their results (see Sect. **??**). It is embedded in the overall functional safety process according to ISO 26262. The created documentation is an essential part for the subsequent steps that result in the safety case. The safety case is the argument that the safety requirements for an item are complete and satisfied by evidence compiled from documents of all ISO 26262 safety activities during the whole lifecycle. It represents the key argument for the Functional Safety Assessment and product release and concludes the ISO 26262 development process.

Aiming at tool support, we started to develop a UML profile and a set of OCL constraints to support the development activities.

The whole approach was presented on the automotive industry conferences VDA Automotive SYS Conference [2], Baden-Baden Spezial 2012 [3] and Safetronic 2014 [4]. The Electronic Steering Column Lock case study is used in all papers and presentations.

In these papers, we introduced (among others) the following stereotypes (see Fig. 2):

---

[2]Presentation on 2012-06-18/20, 2012, Berlin: `http://vda-qmc.de/en/software-processes/vda-automotive-sys/`

[3]2012-10-10/11, Baden-Baden: `http://www.vdi.de/technik/fachthemen/fahrzeug-und-verkehrstechnik/artikel/pressegespraech-auf-der-vdi-tagung-baden-baden-spezial-2012/`

[4]2014-11-11/12 Stuttgart: `https://www.hanser-tagungen.de/web/index.asp?task=001&vid=201402241659596`

- To represent the system to be built the stereotype ≪*Item*≫ is introduced,
- Relevant entities in the environment of the item are called domains (≪*domain*≫),
- Requirements (≪*Requirement*≫) extending UML classes with the an attribute for the requirement text,
- safety requirements (≪*SafetyRequirement*≫) being special requirements with attributes for the ASIL and the safe state,
- safety goals (≪*SafetyGoal*≫) as a top-level requirement being a special safety requirement,
- functional safety requirements (≪*FunctionalSafetyRequirement*≫), also being special safety requirements, systematically derived from the safety goals,
- technical safety requirements (≪*TechnicalSafetyRequirement*≫), also being special safety requirements, systematically derived from the functional safety requirements and being the input for the supplier,
- components or subsystems (≪*CompSubsystem*≫) extending UML classes, and
- to show the relation between technical safety requirements and components or subsystems, the ≪*refersTo*≫-dependency was created.

2.4. Modeling

The implementation of Ford's approach to realize an ISO 26262 compliant safety process (see Sect. 2.3) started off as a document-driven approach using Microsoft products, such as Word, Excel and Visio. The experiences with this approach were good. However, with the growing number of projects using the approach and with increasing complexity of certain features, it is a rather tedious task to keep the different documents consistent and correct amongst each other. Basically, independent documents are created and data is copied manually between the different documents. It is possible to some extent to embedded data or to use Visual Basic for Application (VBA) to provide some means to link data from one document to another. Unfortunately, not everything can be implemented using embedded data and it might not always be possible to use VBA due to corporate regulations. Therefore, it is desirable to move away from a purely document-driven approach. We suggest to use a model-driven approach. With such an approach it is possible to benefit from a global data model allowing different views on this model. Furthermore, it is possible to incorporate the experiences and feedback from the document-driven approach into the envisioned model-driven process. We propose UML [6]. UML is a well-established modeling standard providing a variety of structural and behavioral models with related diagram types. It also offers the concept of stereotypes. Stereotypes give a specific meaning to the element(s) they are attached to. UML already offers profiles with pre-existing stereotypes. However, it is possible to provide additional stereotypes to meet ones needs. This is usually done by providing a new profile containing the additionally defined stereotypes. This profile can then be applied to the model and the additional stereotypes can be used.

For our different method steps, we require stereotypes that are not pre-existing. Therefore, we created profiles that hold all necessary stereotypes relevant to our method. An example for such a stereotype definition is shown in Fig. **??**. In the graphical representation, i.e., the diagram, a stereotype is denoted by ≪*stereotype_name*≫, where stereotype_name denotes the corresponding type. For example, 3WTC in Fig. 1 has the stereotype item (denoted by ≪*item*≫) assigned, identifying it as the system to be build.

Another benefit of a model-driven approach based on UML is that it is possible to provide constraints, e.g., by using the Object-Constraint-Language (OCL) [7], on a model. This way, it is possible to specify syntactic and semantic checks. We specified OCL constraints for all our steps. An example for such an OCL constraint is given in Listing 1.

```
1 Dependency.allInstances()->select(getAppliedStereotypes().name
2   ->includes('realizes')) ->forAll(f|
3       (source.getAppliedStereotypes().name->includes('SubsysComp')) and
4       (target.getAppliedStereotypes().name->includes('LogicalElement')))
```

Listing 1: Validation Condition 1M02LC

It checks that subsystems/components realize logical elements. To perform the check, it is necessary to first select all (Line 2) dependencies (in Line 1) with the stereotypes ≪*realizes*≫ applied (using the EMF keyword **getAppliedStereotypes** in Line 1). For each of the dependencies matching the stereotype, it must be checked if it points from (using the EMF keyword **source** in Line 3) ≪*SubsysComp*≫ to (using the EMF keyword **target** in Line 4) ≪*LogicalElement*≫. The other validation conditions mentioned in this contribution are implemented in a similar way. However, we provide a short textual description of the purpose of the constraint (see e.g. Tab **??**) instead of the actual OCL expression for the remainder of this work. Throughout Sections 3.1 to **??**, we will introduce the definition of the corresponding stereotypes as well as constraints at the appropriate points in our method. The approach is further enhanced by tool support. Section. 5 provides details on how the modeling approach in this section can be realized in a tool framework.

3. Case Study

In previous works, we used an electronic steering column lock (ESCL) as running example (see [1, 2, 8]). However, in this contribution, we introduce a new example: the three-wheeled-tilting control system (3WTC). 3WTC allows leaning the vehicle into a turn based on steering wheel angle and vehicle speed keeping it in balance. This improves stability at low speed curve driving and maneuverability in general. The system is part of the so called "Tilting three-wheeler", see `https://en.wikipedia.org/wiki/Tilting_three-wheeler`. This is a fictitious example system used for ISO 26262 training within Ford and there is no plan to develop such a system or vehicle. However, this example is selected for didactical reasons because its function is easy to understand and the system allows to explain various aspects of ISO 26262.

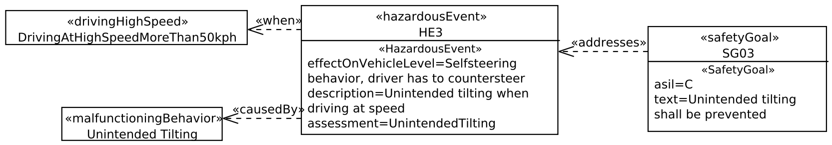

Figure 3: 3WTC Safety Goal including hazardous event, situations and malfunctioning behavior

3.1. Hazard Analysis and Risk Assessment (HARA)

As ISO 26262 is a risk-based functional safety standard, identifying hazards is a vital aspect. Therefore, we start our approach with identifying and classifying potential hazards of the item as described in [1]. In the following paragraphs, we apply the method on the 3WTC example.

1. Provide an Item Definition. ISO 26262 demands a definition of the item, its basic functionality, and its environment. As mentioned in Sect. 2.2, we use a context diagram to represent the item and the domains surrounding it. Figure 1 depicts the context diagram for 3WTC. It contains 3WTC as the item, as well as all relevant domains, e.g., driver, tilt actuator, to achieve tilting of the vehicle upon request. The function, we will further consider in our contribution is *Tilting*.

2. Instantiate Guide-Words. For the 3WTC example, we only consider the malfunctioning behavior *no tilting* and *unintended tilting*. A class with the stereotype ≪*MalfunctioningBehavior*≫ is used to describe any behavior that can be considered as a malfunction of the item. This class has a property *type: MFType*, to link malfunctioning behavior and guide word to each other.

3. Situation Classification. Fig. 3 provides relevant situations for our case study (e.g., ≪*DrivingAtHighSpeedOnNarrowRoads*≫, ≪*DrivingHighSpeed*≫).

4. Hazard Identification. For our example, the combination of *unintended tilting* and *driving at speed* was chosen as an example for a hazardous event (see HE3 in Fig. 3). The effect on the vehicle level, i.e., the effect that can be observed by the driver, is a selfsteering behavior (see property 'effectOnVehicleLevel' in HE3).

5. Hazard Classification by Severity, Exposure, and Controllability. The objective of the hazard classification is to assess the level of risk reduction required for the hazardous event. We executed this step for the hazardous event HE3 from our 3WTC example. Figure 4 captures our results of the risk assessment for HE3 given in Fig. 3. With the rating of S3, E4, and C2, we obtain an ASIL C.

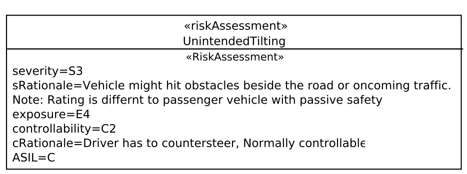

Figure 4: Risk Assessment for one Hazardous Event of 3WTC

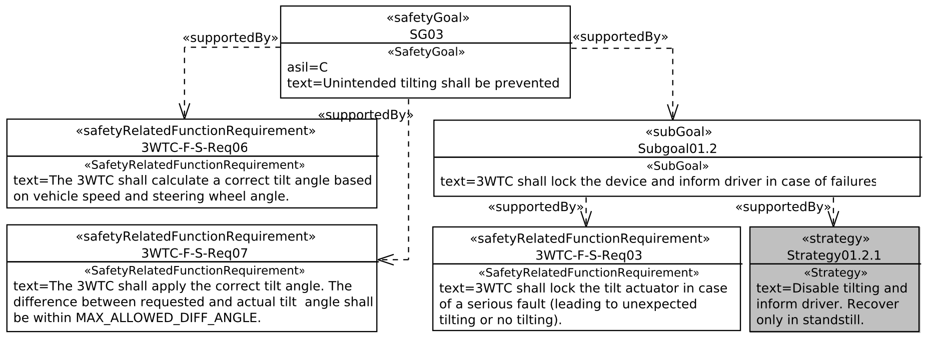

Figure 5: 3WTC Goal Structure for SG03

6. Define and Verify Safety Goals. To address the hazardous event, we derived the safety goal "*Unintended tilting shall be prevented.*" The safety goal is given in Fig. 3, right-hand side. The figure also provides the relations between safety goal, hazardous events, situations, and malfunctioning behavior.

3.2. Functional Safety Concept (FSC)

After the hazard analysis and risk assessment, the next step is to break down the high-level safety goals into functional safety requirements and allocate them to logical elements of a preliminary architecture as described in [2].

1. Break-down safety goals into functional safety requirements. Figure 5 illustrates the goal structure for deriving functional safety requirements for the safety goal obtained in Sect. 3.1 for the 3WTC example. For this particular safety goal, we derived a set of functional safety requirements. The naming convention we used is `Feature abbreviation`-F-S-Req `running number`. In Fig. 6, we show the warning and recovery concept (W&R) related to SG03. It starts off, where *Strategy01.2.1* given in Fig. 5 stopped. For the warning and recover concept, an additional two functional safety requirements have been derived. The first one (3WTC-F-S-Req04) deals with the concept of driver information and the second one (3WTC-F-S-Req05) with necessary recovery conditions.

2. Specify all applicable attributes of the requirements. To illustrate our approach, we select 3WTC-F-S-Req06 (see upper left-hand side of Fig. 5) as a representative of a

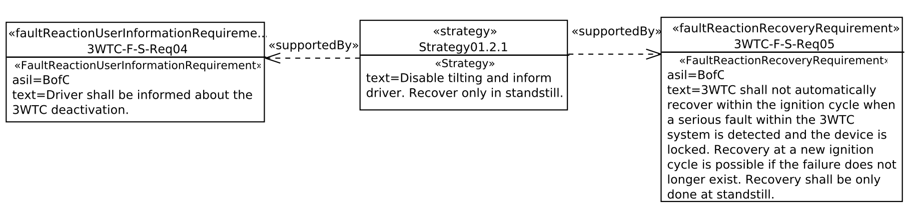

Figure 6: 3WTC Warning and recovery Concept for SG03

Safety Req-ID	3WTC-F-S-Req06	**Strategy/Subgoal**	01.2 (subgoal)/01.2.1 (strategy)
Safety Goal Ref.	SG03	**Operating Modes**	3WTC Normal Operation
ASIL Classification (if applicable)	C	**Safe State** (if applicable)	No tilting
Functional Safety Requirement	The 3WTC shall calculate a correct tilt angle based on vehicle speed and steering wheel angle.		
Purpose	To prevent steering column locking while vehicle is moving at speed and steering is required.		
Fault Tolerant Time interval (if applicable)	200ms		
Reduced Functionality interval (if applicable)	n/a		
Functional Redundancies (e.g. fault tolerance) (if applicable)	n/a		
Description of actions of the driver or other endangered persons (if applicable)	n/a		
Validation Criteria for these actions (if applicable)	n/a		
V&V method	Design and methods review		
V&V acceptance criteria	Design and methods are appropriate for required ASIL.		

Table 1: 3WTC Attributes for 3WTC-F-S-Req06

safety related function requirement. The attributes, we must provide for this category are fault tolerant time (ftt), emergency operation interval (emergencyOpInterval), description of driver or other involved persons action (descriptionOtherPersonsAction), and validation criteria for the aforementioned actions (validationCriteriaForActions). As a safety related function is also a functional safety requirement, the following attributes have to be provided, as well:

- related safety goal,sub-goal, strategy, *(These three attributes can be looked up in the related goal structure.)*
- operating modes, *(The related requirement is only valid for a given set of operating modes. Usually, some indication on the operating modes is given in the item definition)*
- purpose, *(The purpose of a safety requirement may be similar to the strategy or sub-goal if any exist.)*
- verification and validation method, *(An example for such a method could be testing.)*
- acceptance criteria considering verification and validation, *(An example for such criteria could be that all test cases pass.)*

3. Check for completeness of defined requirements. In our contribution, we consider only one safe state, namely *No tilting*. This safe state is covered by safety-related function 3WTC-F-S-Req06. For the assumptions *A1.1 Balance point is between wheels* and *A3.1 Tilting is only active during forward driving* general requirements 3WTC-F-S-Req10 and 3WTC-F-S-Req11 (not shown in this contribution) exist. For safe state *No tilting*, user information is covered by 3WTC-F-S-Req04, and recovery is covered by 3WTC-F-S-Req05 The only operating mode considered in this contribution is *3WTC Normal Operation*. This operating mode is referred to by 3WTC-F-S-Req01 – 3WTC-F-S-Req07. Within the scope set in this contribution, the investigation of requirements

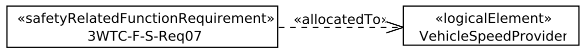

Figure 7: 3WTC Requirement Allocation

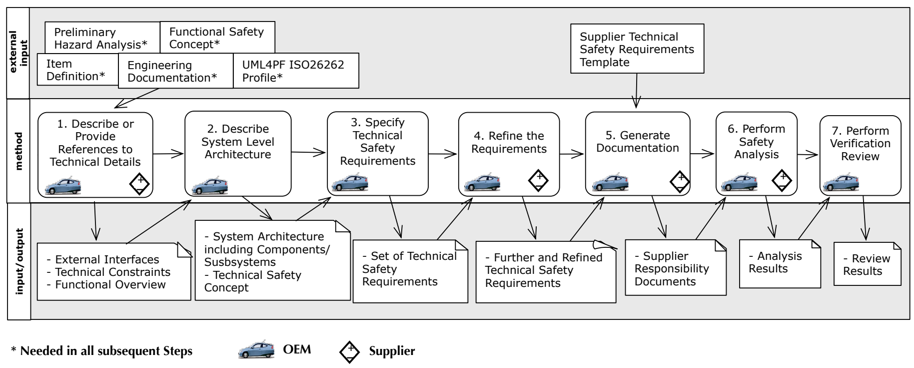

Figure 8: Technical Safety Requirements Specification Method considering the OEM/Supplier Interface

necessary to ensure controllability referring to technical means or controls necessary for driver (or other persons involved) actions, no additional requirements have been identified.

4. ASIL decomposition. For our selected functional safety requirement 3WTC-F-S-Req06, no ASIL decomposition is necessary.

5. Allocation of Requirements. For our selected example, one requirement allocation is given in Fig. 7.

6. Safety Analysis, Simulation, and Test. For our 3WTC example, the goal structures provided in Figs. 5 and 6 are sufficient qualitative analysis to show that the functional safety requirements are consistent and compliant to the safety goals and are able to mitigate or avoid the hazardous events. Simulation and tests are performed to check the controllability assumptions. However, the results of these analyses are not given in this contribution.

4. Technical Safety Requirements Specification (SRS) Method

The aim of the analysis is to specify technical safety requirements according to the technical safety concept and the allocation of the functional safety requirements to logical elements of the preliminary architecture. Figure 8 depicts an overview of our method. We highlight for each activity the contribution of the OEM and its supplier.

Step 1. Describe or Provide References to Technical Details. The OEM provides the majority of information for this step and requests specific documentations of interfaces of components a supplier constructed. The supplier is just reacting upon demand of the OEM and has no active role in this step. The reason is that the OEM is responsible for the overall system and has the necessary overview to describe or demand descriptions of all parts.

We create safety requirements specifications describing how the safety measures located in the functional safety concept should be implemented and update the hazard analysis and risk assessment in case we identified new hazards or situations.

To derive the safety requirements specifications, we proceed as follows:

- *Describe or provide reference to details of external interfaces of the item.* The description from the item definition can be used and refined by specifying all parameters of the signals in detail.
- *Describe or provide reference to technical constraints.* Technical constraints are functionalities that are implemented in the same way for all vehicles.
- *Describe a functional overview of components/subsystems contained in the item.* Furthermore, describe a clear boundary of the item and its surroundings. State the main task and purpose for all elements located outside of the item boundary. For each component/subsystem the highest ASIL of the allocated functional safety requirements (for more details see [2]) is documented. The logical elements of our preliminary architecture are mapped to components/subsystems.

As a representative of the stereotype we introduced for this step, we selected ≪*SubsysComp*≫ (see Figure 10).

In the first step, we set the attributes *description*, *inside*, and *asil*. Figure 9 (center) shows these attributes for the relevant subcomponent Speed Sensor Modul (SSM). The *description* gives an overview on the realized functionality. Note that the property *inside* illustrates whether the component is inside the system boundary of the item. This information can usually be found in the item definition. The ASIL is set to the highest ASIL of the requirements referring to the subsystem or component.

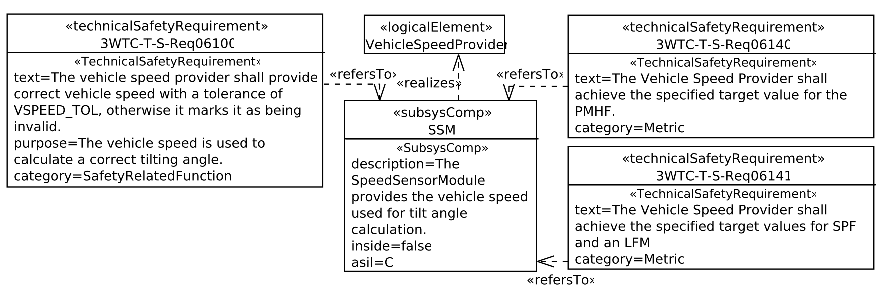

Figure 9: 3WTC SRS Elements

Step 2. Describe System Level Architecture. The OEM describes the system architecture. This is an OEM task because the architecture requires complete information about the technical details. Any information required from the supplier should be gathered in the previous step.

The input is used to set up a system level architecture. This architecture may be represented, for example, as a UML composite diagram. The architecture in this step is enriched by a technical safety concept (e.g. redundancy) for every safety goal with

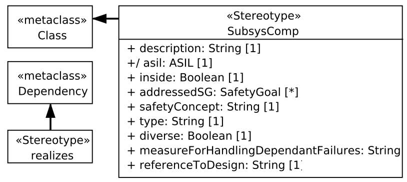

Figure 10: Profile Part concerning (Sub-) Components

Step	ID	Condition
1	1M01DE	The description of components/subsystems is not allowed to be empty. In particular, each class with the stereotype ≪*SubsysComp*≫ must have an attribute 'description: String'.
1	1M02LC	Subsystems or components realize logical elements. A ≪*realizes*≫ stereotype is attached to a dependency from a class with the stereotype ≪*SubsysComp*≫ to a class with the stereotype ≪*LogicalElement*≫.

Table 2: SRS: Validation Conditions for Step 1 (excerpt)

an ASIL rating higher than ASIL B. Whenever redundancy is used, we are required to provide the type of redundancy (e.g. HW or SW). In addition, it is necessary to clarify if it is a diverse or homogeneous redundancy. In both cases, measures for handling potential dependent failures must be described.

In this step, the attributes *safetyConcept*, *type*, *diverse*, and *measureForHandlingDependantFailures* of ≪*SubsysComp*≫ have to be provided. For the subsystem component relevant to our 3WTC example, these values are set in the same way as those previously described.

Table 3 contains an excerpt of checks for this step.

2	2C01SG	Every safety goal has to be realized by at least one component/subsystem.
2	2C02DR	If a component realizes a safety goal with ASIL greater than ASIL B, a concept for redundancy shall be defined.

Table 3: SRS: Validation Conditions for Step 2(excerpt)

Step 3. Specify Technical Safety Requirements. The OEM describes the OEM specific parts of the technical safety requirements. This is an OEM task, because the OEM has the knowledge of the overall architecture, while the supplier knows isolated parts and cannot elicit technical safety requirements for parts unknown to it and in particular consider consequences of the interactions of known components with unknown components.

Generally speaking, the task of this step is to split the functional safety requirements up into technical safety requirements. To do this, we start with the functional safety requirement and the components or subsystems that realize this requirement. To find out which component or subsystems realize the functional safety requirement, the mapping from logical elements to components or subsystems is used. For the relevant elements of 3WTC, this mapping is shown in Fig. 9. For each component, the part of the functional requirement that should be realized, as well as its requirement text is described. For each technical safety requirement, a unique ID, the reference to the

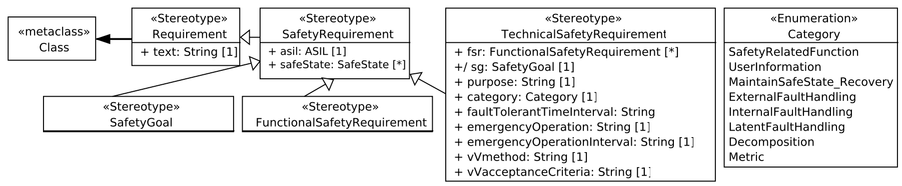

Figure 11: Profile Part concerning Safety Requirements

functional safety requirement it realizes, as well as the component or subsystem it is assigned to, is specified. The ASIL is derived from the ASIL of the functional safety requirement. Summarized, the following aspects have to be captured according to [3, Part 4, 6.4.2]:

- Reference to the functional safety requirement (FSR),
- Reference to the component/subsystem,
- Unique ID,
- ASIL (derived from the ASIL of the functional safety requirement),
- Technical safety requirement text,
- Purpose of the requirement,
- Safe state, and
- Category

The right-hand side of Fig. 11 contains all currently identified categories. For each functional safety requirement, we go through every category entry and decide whether it is relevant for the respective functional safety requirement. For those considered relevant, we fill out the corresponding template. Note that requirements of some categories (e.g., 'Decomposition' or 'Metric') may be defined at a later point.

Figure 9 shows three examples of technical safety requirements for our 3WTC example. For technical safety requirement 3WTC-T-S-Req061000, a subset of the just mentioned attributes is given.

Table 4 provides an excerpt of consistency checks relevant to this step.

Step	ID	Condition
3	3M01ID	Technical safety requirements have a reference to a component/subsystem and a unique ID is set.
3	3M02RA	Requirement text, purpose, and safe state have to be defined for all technical safety requirements.

Table 4: SRS: Validation Conditions for Step 3 (excerpt)

Step 4. Refine Requirements. The OEM refines the OEM specific parts of the technical safety requirements. This is an OEM task, because the OEM has the knowledge of the overall architecture, while the supplier knows isolated parts and cannot elicit technical safety requirements for parts unknown to it and in particular consider consequences of the interactions of known components with unknown components. Afterwards, the supplier is contacted to agree on these requirements.

At this place, the technical safety requirements of the previous step are investigated in more detail. The following activities have to be conducted:

- *Decomposition with independence argumentation*. For details on this topic, please refer to Part 8 of ISO 26262.
- *Hardware metric derivation and rationale*. Hardware metrics - as required by ISO 26262 part 5 - are derived and the break-down to components/subsystems is justified. This break-down of metric requirements enables a distributed development and is necessary to have a clear OEM/Supplier interface. The Maximum Probability of Safety Goal violation due to random Hardware Failures (PMHF) has to be achieved on safety goal level, i.e. by all components contributing to the Safety Goal. The PMHF value for SG03 has to be split into separated target values for the Steering Wheel Angle Provider, the Vehicle Speed Provider and the TiltActuator. In order to obtain the different target values, we first need to assign an initial value to the PMHF in question. We use the initial values to perform a fault tree analysis. Based on the outcome of this analysis, we can assign or adjust the PMHF for the respective module. The target value for the Vehicle Speed Provider is inserted into the refined requirement 3WTC-T-S-Req07141. If redundancy concepts are applied and the fault detection is not limited to a single component, target values for Single Point Fault Metric (SPFM) and the Latent Fault Metric (LFM) have to be derived for each component. This calculation is based on the target values of the Safety Goal as given by ISO 26262. Otherwise, the SPFM and the LFM of the Safety Goal can be directly cascaded to all components that realize requirements derived from that Safety Goal.
- *Elicitation of requirements concerning the ability to configure a system by calibration data*. For details on this topic, please refer to the corresponding part of ISO 26262.
- *Identify Parameters used in several requirements*. For these parameters, boundary values should be defined. In the example, we refine 3WTC-T-S-Req06100. It makes use of the parameter "VSPEED_TOL", representing the allowed tolerance of the vehicle speed value. For this parameter, we define a preliminary value needed for the correct calculation of the tilt angle. The constraint considered is that the upper boundary of the range is not hazardous.
- *Specify requirements for operation, service and decommissioning*. For details on this topic, please refer to the corresponding section of ISO 26262.

Within the tool, it is necessary to complete the properties which have been postponed in the previous step.

Table 6 shows the content inserted into the stereotype attributes for one technical safety requirement.

Table 5 introduces an excerpt of consistency checks.

Step 5. Generate Documentation. The OEM generates the initial set of documents that are presented in form of a template, which the supplier has to instantiate.

The OEM provides the content defined in the previous steps and the supplier adds the details, because the supplier has the knowledge of its components and the ability to perform the safety analysis for the component. The template is precise about which

Step	ID	Condition
4	4C01AF	The ASIL of the technical safety requirement is consistent to the ASIL in the corresponding functional safety requirement.
4	4G02FF	Fault tolerant time interval is consistent with the corresponding functional safety requirement.

Table 5: SRS: Validation Conditions for Step 4 (excerpt)

T-S-Req-ID	3WTC-T-S-Req06100
Safety Goal(s)	SG01, SG02, SG03
FSR	3WTC-F-S-Req06, 3WTC-F-S-Req01, 3WTC-F-S-Req02
ASIL	C
Safe State	SSM quality factor is set to invalid or no vehicle speed signal is provided
TSR Text	The vehicle speed provider shall provide correct vehicle speed with a tolerance of VSPEED_TOL otherwise it marks it as being invalid.
Purpose	The vehicle speed is used to calculate a correct tilting angle.
Category	Safety Related Function Requirement
V&V Method	Review design and methods review at supplier. Vehicle test at all speed ranges. Fault insertion in sensor.
V&V Acceptance Criteria	Design and method are appropriate for required ASIL. Correct vehicle speed is delivered. Faults lead to quality flag = invalid.

Table 6: Generated Technical Safety Requirements

details are needed and reduces discussions and the risk of missing information in the overall safety analysis performed in the next step.

Based on the technical safety requirements, a document is generated for each relevant component/subsystem. These documents detail the supplier's responsibilities.

Table 6 shows the table generated from the model for one technical safety requirement.

The component/subsystem provider has to define the architecture / redundancy concept including:

- A description of the architecture / redundancy concept
- The type of redundancy, e.g. information redundancy, time redundancy, hardware redundancy or software redundancy, including a justification why it is suitable
- A statement if diverse or homogeneous redundancy is used
- A description of measures for handling potential dependent failures

Furthermore, they have to define the latent fault handling including:

- Measures related to the detection and indication of faults in the component itself
- Avoidance of latent faults
- Multiple point fault detection interval
- Details on fault reaction

This information has to be made available for review purposes. Further relevant documents have to be referenced, as well.

Step	ID	Condition
5	5G01DC	Generate supplier documentation including purpose of each component, requirements for the component or subsystem, and a list of aspects to be completed by the supplier.

Table 7: SRS: Validation Conditions for Step 5 (excerpt)

Step 6. Perform Safety Analysis. Based on the documentation generated so far, the OEM performs a safety analysis. Note that the OEM asks the supplier for a safety analysis of subsystems that the supplier builds alone. The OEM conducts the safety analysis of the overall system without the supplier, because only the OEM has the knowledge of the overall system and all details provided by suppliers.

To perform the safety analysis, a reference to the design of components/subsystem should be given. The safety analysis shows compliance and consistency between the technical safety concept with its technical requirement, the functional safety concept, and the preliminary architecture. An analysis shall also verify the system design regarding compliance and completeness with regard to the technical safety concept. This is why the description of components/subsystems in the respective stereotype ≪*SubsysComp*≫ has an attribute ’referenceToDesign: String’.

The safety analysis is performed using a structured fault tree. This fault tree will be subject of a planned publication.

Step	ID	Condition
6	6C01RD	For each components/subsystems, the attribute referenceToDesign is not empty.
6	6SI01DE	Description of components/subsystems (≪*SubsysComp*≫ has attribute ’referenceToDesign: String’)

Table 8: SRS: Validation Conditions for Step 6 (excerpt)

Step 7. Perform Verification Review. ISO 26262 requires to perform a verification review of the functional safety concept by a different person than the author of the review and a person who knows the technology of the system under development. This is supported by OCL validation constraints and the generation of a structured document from the model. The OEM performs the verification review without the supplier, due to its overall responsibility of the system. At this point in time the OEM should have gathered all required technical details in the previous steps of our method to conduct the verification review alone.

5. Tool Support

In sect.2.4, we stated how the previously document-driven approach could be transfered to a model-driven one. We now describe how this model-driven approach can be fitted with tool-support. When deciding on tool-support, one has to decide whether to develop a new tool or to use an existing one and adapt it. In our case, we used the latter approach.
We use a tool called UML4PF, developed at the University of Duisburg-Essen, and integrated support for FIFS as described in Sects. 3.1 – **??** into it. UML4PF is based on

the Eclipse platform [9] together with its plug-ins EMF [10] and OCL [7]. Our UML-profiles are conceived as an Eclipse plug-in, extending the EMF meta-model. The OCL constraints are integrated directly into the profile. Thus, it is possible to automatically check the constraints using the validation mechanisms provided by Eclipse.

After the developer has drawn some diagram(s) using an EMF-based editor, for example Papyrus UML [11] and applied our stereotypes, UML4PF provides him or her with the following functionality: it checks if the developed model is valid and consistent by using our OCL constraints (represented textually throughout this contribution). It returns the location of invalid parts of the model, and generates documentation that can be used for the manual validation and review activities.

6. Related Work

HARA. We are not aware of any publications about a structured and model-based hazard analysis and risk assessment for automotive systems equipped with integrity checks.

Two hazard analysis methods are compared by Törner et al. [12]. The paper shows that the adapted functional failure analysis (FFA) is less time-consuming than the method of the European Space Agency (ESA method). The method presented is this paper is based on the results of [12].

The entire safety lifecycle including hazard analysis and risk assessment is presented by Baumgart [13]. Our method can complement the hazard analysis of Baumgart's safety lifecycle.

The Safety Management System and Safety Culture Working Group provides guidance on hazard identification by different means, e.g., brainstorming, HAZOP, checklists, FMEA [14]. Their results are considered in the method presented in this paper.

Jesty et al. [15] give a guideline for the safety analysis of vehicle-based systems, including system analysis, hazard identification, hazard analysis, identification of safety integrity levels, FMEA, and fault tree analysis. Their work also uses the HAZOP guidewords, but they focus on the safety integrity level as defined in the IEC 61508 and not on the ASIL from ISO 26262. Jesty et al. additionally address FMEA and fault tree analysis for analyzing existing systems, but do not consider a model or validation conditions.

In contrast to our work, which focuses on the determination of necessary risk reduction, following papers describe model-based approaches specific for later development phases, when the system is already designed and not the determination of necessary risk reduction:

Papadopoulos and Grante [16] propose a process that addresses both cost and safety concerns and maximizes the potential for automation to address the problem of increasing technological complexity. It combines automated safety analysis with optimization techniques.

Li and Zhang [17] present a comprehensive software hazard analysis method, which applies a number of hazard analysis techniques, and the proposed method is applied to a software development process of a control system. The described method for hazard analysis is similar but less detailed than ours.

Mehrpouyan [18] proposes a model-based hazard analysis procedure (based on SysML models) for the early identification of potential safety issues caused by unexpected environmental factors and subsystem interactions within a complex safety-critical system. The proposed methodology additionally maps hazard and vulnerability modes to specific components in the designed system and analyzes the hazards.

Zhang et al. [19] propose a comprehensive hazard analysis method based on functional models. It mainly addresses fault tree analysis and FMEA.

Giese et al. [20] present an approach that supports the compositional hazard analysis of UML models described by restricted component and deployment diagrams. It also starts with environment models, but then focuses on the safety analysis of the design.

Hauge and Stølen [21] introduce the SaCS method. The method provides guidance on how to select and use patterns for the development of safety control systems. The patterns are categorized into process and product patterns. This work differs from our own, because we focus specifically on early hazard analysis and provide detailed guidance.

FSC. Basir, Denny, and Fischer [22] present goal structures for safety cases in the automotive sector. They do not focus on the technical realization but consider the entire safety process with their documents as entities.

Dittel and Aryus [23] present an overview of V&V activities at Ford Motor Company applied for the lane keeping aid system. This paper also presents elements of the process for functional safety according to ISO 26262, i.e. the analysis activities.

Sinha [24] illustrates an example of a brake-by-wire system for road vehicles including a safety and reliability analysis compliant to ISO 262626. The conclusions derive suggestions for future projects, such as that the system architecture of road vehicles shall support the detection of failures and have the means to still provide desired services until the failures are repaired.

Palin et al. [25] provide guidelines for safety practitioners and researchers to create safety cases compliant to the ISO 26262 standard. The authors propose extensions of the Goal Structuring Notation, patterns, and a number of re-usable safety arguments for creating safety cases. For confidentiality reasons, the authors cannot show example instantiations of their patterns or generic arguments.

Conrad et al. [26] compares software tools that support ISO 26262 certification. The authors identified a list a qualification requirements for selecting ISO 26262 support tools. The publication also contains a report about Conrad et al.'s experience with these tools.

Hillebrand et al. [27] discuss how to develop electric and electronic architectures (EEA) compliant with the ISO 26262 standard. The authors focus on safety requirements during early development phases. Hillenbrand et al. present a method for eliciting safety requirements, and mapping their safety concerns to functions of design artifacts. Previously, Hillebrand et al. [28] proposed a model-based and tool- supported approach for the failure mode and effect analysis (FMEA) of EAAs complaint to ISO 26262. The authors contribute a formalized method for eliciting and analyzing data for a FMEA.

Habli et al. [29] propose a process for model-based assurance for justifying automotive functional safety. They use SysML and GSN as graphical notations. Their goal and ours is similar. We both want to support a method based on ISO 26262 to derive functional safety requirements. In contrast to their work, we use UML, which gives us a broader spectrum of modeling possibilities. Furthermore, we provide tool support for our method and equipped our approach with formal consistency checks on the model. These checks can be automatically checked by our tool. In addition, our way of modeling allows us to trace elements within our models.

Born et al. [30] report on lessons learned from applying a model-based approach for ISO 26262 certification. The authors also discuss the advantages of models instead of text in the ISO 26262 certification process

SRS. We are not aware of any publication about a structured and model-based safety requirements analysis with a focus on the OEM-supplier interface for automotive systems equipped with integrity checks. Chen et al. [31] provide modeling support for ISO 26262 software development. In contrast to our work, the authors focus on providing support for the analysis of malfunctions and the hazards they cause. In particular, the work illustrates how to model errors and error propagation in an automotive system.

Habili et al. [32] show a model-based method for creating a functional safety concept compliant to ISO 26262. The authors extend the SysML modeling notation with new diagram types. Different to our work their approach is limited to functional safety requirements that are elicited based on diagrams. Moreover, they do not provide formal OCL checks nor a structured method.

Tang et al. [33] present an approach for explicitly integrating the supplier into the product lifecycle of automotive development. The authors present a high level process for the entire product lifecycle management, and in contrast to our work do not focus on detailed requirements analysis.

The entire safety lifecycle including safety requirements analysis is presented by Baumgart [13], who also considers the supplier interface. Our method can complement the analysis of Baumgart's safety lifecycle, because we offer a greater level of detail.

The Safety Management System and Safety Culture Working Group provides guidance on functional safety development by different means, e.g., brainstorming, HAZOP, checklists, FMEA [14]. Their work considers also the interface between systems and stakeholders, but does not focus in particular on a supplier interface or the automotive industry.

Jesty et al. [15] give a guideline for the safety analysis of vehicle-based systems, including system analysis, hazard identification, hazard analysis, identification of safety integrity levels, FMEA, and fault tree analysis. They focus on the safety integrity level as defined in the IEC 61508 and not on ASIL from ISO 26262. Jesty et al. do not consider a model or validation conditions and do not focus on the supplier interface.

In contrast to our work, who focuses on the safety requirements analysis concerning the supplier interface, the following papers describe model-based approaches specific for later development phases, when the system is already designed and not the determination of necessary risk reduction:

Papadopoulos and Grante [16] propose a process that addresses both cost and safety concerns and maximizes the potential for automation to address the problem of increasing technological complexity. It combines automated safety analysis with optimization techniques.

Giese et al. [20] present an approach that supports the compositional hazard analysis of UML models described by restricted component and deployment diagrams. It also starts with environment models, but then focuses on the safety analysis of the design and does not focus on the supplier interface.

V&V. We are not aware of any publication about a model-based structured validation and verification of automotive systems with a focus on the OEM-supplier interface for automotive systems equipped with integrity checks. Maropoulos et al. [34] presented a survey of industrial verification and validation efforts. The report presents evidence that verification and validation of products and processes is vital for complex products and in particular modeling and planning of such methods are an ongoing research challenge. Sinz et al. [35] used formal methods to validate automotive product configuration data. In contrast to our work, their method specifically focuses on detecting inconsistencies in product configurations of vehicles to support business decisions. Instead we focus on technical verification and validation efforts. Bringman et al. [36] described the impact model-driven design has in the automotive industry and showed how models can be used to derive test cases during different steps of the automotive product lifecycle. In contrast to our work Bringman et al. focus exclusively on model-based testing of automotive systems. Dubois et al. [37] presented a method for model-based validation and verification efforts to check if the final product matches initial requirements. In contrast to our work Dubois et al. focus on using UML-based models to create test cases for more detailed implementation models in e.g. SIMULINK. Montevechi et al. [38] focuses on the simulation of processes in the automotive industry. Their methodology builds simulation models to analyze which combinations of variables can lead to problems. Within the automotive industry, different activities are started to extend the safety processes with model-based system engineering aspects, mainly focusing on architecture description[5] and semiautomatic safety analyses [39].

7. Conclusion

Our method has been applied to several Ford of Europe projects. However, the formal validation conditions and tool support was not used in these projects and was developed as contribution for this paper. We are confident that this contribution will ensure the same consistency and correctness of future verification & validation with less effort than the manual approach currently used. The main contributions of our approach are:

The main contribution of our approach is a Structured Method helping to are:

- select relevant situations from the hierarchically organized profile for the hazard analysis to reduce the risk of forgetting a relevant situation,

[5]Electronics Architecture and Software Technology - Architecture Description Language, http://www.east-adl.info/

- ensure that only situations are considered that are relevant for the function in question,
- describe the effect of a malfunction on system and on vehicle level to make the hazard analysis comprehensible for different stakeholders and enable an efficient team verification of the hazard analysis,
- structure the analysis in different steps on different levels and foster an alignment between the analysis and the organizations (departments with experts regarding hardware/ software, system level, vehicle/functional level) involved in the creation and review of the analysis,
- support the definition of safety goal definitions suitable to derive the system design,
- derive functional safety concepts for the automotive domain compliant to ISO 26262,
- ensure consistency between the safety requirements, safety analyses and safety V&V,
- define a complete set of V&V activities, including reviews, analyses, simulations and tests by using pre-defined V&V activities based on the category of the requirement,
- allocate the V&V activities between OEM and the involved suppliers,
- define due dates,
- collect and assess the V&V results for all requirements, and
- provide input to the safety case.

In this paper, we describe the overall process and add a structured method for requirements management, helping to

- define the interface to the suppliers and address functional safety,
- break down the functional safety requirements into technical safety requirements,
- perform a metric breakdown,
- ensure the completeness of technical safety requirements by using tables with predefined cells.

Our UML profile contains all relevant elements for a hazard analysis, functional safety concept, technical safety requirements specification and safety V&V. The UML profile provides the basis for creating a model for the safety development in compliance with ISO 26262. Thus, we provide a computer-aided technique to discover errors in the complete safety development process caused by inconsistencies or errors in one or more (UML) diagrams.

The safety development documents, including the supplier interface, in practice are currently document based using spreadsheet-processing tools from Microsoft Office. We propose to conduct the analysis on UML models and to create tables from the models for the different artifacts. Thus, we use a model-based approach, but the suppliers will receive the same type of documentation they are used to.

In the future, we will extend the approach to Safety Analysis and Safety Management. Currently, Ford is implementing tool support in NoMagics MagicDraw. Ford is also creating import and export functionality for their current templates and is developing an interface to requirements management tools.

References

[1] K. Beckers, T. Frese, D. Hatebur, M. Heisel, A Structured and Model-Based Hazard Analysis and Risk Assessment Method for Automotive Systems, in: Proc. of the 24th IEEE Int. Symposium on Software Reliability Engineering, IEEE, 238–247, URL `http://www.ieee.org/`, 2013.

[2] K. Beckers, I. Côté, T. Frese, D. Hatebur, M. Heisel, Systematic Derivation of Functional Safety Requirements for Automotive Systems, in: Proceedings of SAFECOMP, LNCS 8666, Springer, 65–80, 2014.

[3] International Organization for Standardization (ISO), Road Vehicles – Functional Safety, ISO 26262, 2011.

[4] International Electrotechnical Commission (IEC), Functional safety of electrical/-electronic/programmable electronic safety-relevant systems, IEC 61508, 2000.

[5] M. Jackson, Problem Frames. Analyzing and structuring software development problems, Addison-Wesley, 2001.

[6] UML Revision Task Force, OMG Unified Modeling Language: Superstructure, Object Management Group (OMG), 2010.

[7] UML Revision Task Force, OMG Object Constraint Language: Reference, URL `http://www.omg.org/docs/formal/10-02-02.pdf`, 2010.

[8] K. Beckers, I. Côté, T. Frese, D. Hatebur, M. Heisel, A Structured Validation and Verification Method for Automotive Systems considering the OEM/Supplier Interface Technical Report, Tech. Rep., https://www.uni-due.de/imperia/md/content/swe/papers/vav2015tr.pdf, 2015.

[9] Eclipse Foundation, Eclipse - Development Platform, http://www.eclipse.org/, 2011.

[10] Eclipse Foundation, Eclipse Modeling Framework Project (EMF), http://www.eclipse.org/modeling/emf/, 2012.

[11] Atos Origin, Papyrus UML Modelling Tool, http://www.papyrusuml.org/, 2011.

[12] F. Törner, P. Johannessen, P. Öhman, Evaluation of Hazard Identification Methods in the Automotive Domain, in: J. Górski (Ed.), SAFECOMP 2006, LNCS 4166, Springer, 237–260, 2006.

[13] S. Baumgart, Investigations on Hazard Analysis Techniques for Safety Critical Product Lines, in: IRSCE12, ACM, 2012.

[14] Safety Management System and Safety Culture Working Group (SMS WG), Guidance on hazard identification, Tech. Rep., 2009.

[15] P. H. Jesty, K. M. Hobley, R. Evans, I. Kendal, Safety analysis of vehicle-based systems, in: Proc. of the 8th Safety-critical Systems Symposium, LNCS 1943, Springer, 90–110, 2000.

[16] Y. Papadopoulos, C. Grante, Evolving car designs using model-based automated safety analysis and optimisation techniques, Journal of Systems and Software 76 (1) (2005) 77 – 89.

[17] W. Li, H. Zhang, A software hazard analysis method for automotive control system, IEEE Computer Society, 744–748, 2011.

[18] H. Mehrpouyan, Model-Based Hazard Analysis of Undesirable Environmental and Components Interaction, Master's thesis, Linköpings Universitet, 2011.

[19] H. Zhang, W. Li, W. Chen, Model-based hazard analysis method on automotive programmable electronic system, in: 3rd International Conference on Biomedical Engineering and Informatics (BMEI), 2658–2661, 2010.

[20] H. Giese, M. Tichy, D. Schilling, Compositional Hazard Analysis of UML Component and Deployment Models, in: SAFECOMP, LNCS 3219, Springer, 166–179, 2004.

[21] A. A. Hauge, K. Stølen, A Pattern-Based Method for Safe Control Systems Exemplified within Nuclear Power Production, in: SAFECOMP, LNCS 7612, Springer, 13–24, 2012.

[22] N. Basir, E. Denney, B. Fischer, Deriving Safety Cases for Hierarchical Structure in Model-Based Development, in: SAFECOMP 2010, LNCS 6351, Springer, 68–81, 2010.

[23] T. Dittel, H.-J. Aryus, How to 'Survive' A Safety Case According to ISO 26262, in: SAFECOMP 2010, LNCS 6351, Springer, 97–111, 2010.

[24] P. Sinha, Architectural design and reliability analysis of a fail-operational brake-by-wire system from ISO 26262 perspectives, Reliability Engineering & System Safety (2011) 1349 – 1359ISSN 0951-8320, doi:\bibinfo{doi}{http://dx.doi.org/10.1016/j.ress.2011.03.013}, URL `http://www.sciencedirect.com/science/article/pii/S095183201100041X`.

[25] R. Palin, D. Ward, I. Habli, R. Rivett, ISO 26262 safety cases: Compliance and assurance, in: System Safety, 2011 6th IET Int. Conf. on, 1–6, 2011.

[26] M. Conrad, P. Munier, F. Rauch, Qualifying software tools according to ISO 26262, in: Proc. Dagstuhl-Workshop Modellbasierte Entwicklung eingebetteter Systeme (MBEES10), 2010.

[27] J. Hillebrand, P. Reichenpfader, I. Mandic, H. Siegl, C. Peer, Establishing Confidence in the Usage of Software Tools in Context of ISO 26262, in: Computer Safety, Reliability, and Security, LNCS, Springer, 257–269, 2011.

[28] M. Hillenbrand, M. Heinz, N. Adler, J. Matheis, K. Müller-Glaser, Failure mode and effect analysis based on electric and electronic architectures of vehicles to support the safety lifecycle ISO/DIS 26262, in: Rapid System Prototyping (RSP), 2010 21st IEEE International Symposium on, 1–7, 2010.

[29] I. Habli, I. Ibarra, R. Rivett, T. Kelly, Model-Based Assurance for Justifying Automotive Functional Safety, in: SAE Technical Paper 2010-01-0209, doi: \bibinfo{doi}{10.4271/2010-01-0209}, 2010.

[30] M. Born, J. Favaro, O. Kath, Application of ISO DIS 26262 in Practice, in: Procs of the 1st Workshop on Critical Automotive Applications: Robustness & Safety, CARS '10, ACM, New York, NY, USA, 3–6, 2010.

[31] D. Chen, R. Johansson, H. Lönn, Y. Papadopoulos, A. Sandberg, F. Törner, M. Törngren, Modelling Support for Design of Safety-Critical Automotive Embedded Systems, in: Computer Safety, Reliability, and Security, vol. 5219 of *LNCS*, Springer Berlin Heidelberg, 72–85, 2008.

[32] I. Habli, I. Ibarra, R. Rivett, T. Kelly, Model-Based Assurance for Justifying Automotive Functional Safety, in: SAE World Congress, Springer Berlin Heidelberg, 1–16, 2010.

[33] D. Tang, X. Qian, Product lifecycle management for automotive development focusing on supplier integration, Computers in Industry 59 (23) (2008) 288 – 295.

[34] P. G. Maropoulos, D. Ceglarek, Design verification and validation in product lifecycle, CIRP Annals - Manufacturing Technology 59 (2) (2010) 740–759.

[35] C. Sinz, A. Kaiser, W. Küchlin, Formal Methods for the Validation of Automotive Product Configuration Data, Artif. Intell. Eng. Des. Anal. Manuf. 17 (1) (2003) 75–97.

[36] E. Bringmann, A. Kramer, Model-Based Testing of Automotive Systems, in: Software Testing, Verification, and Validation, 2008 1st International Conference on, 485–493, 2008.

[37] H. Dubois, M. Peraldi-Frati, F. Lakhal, A Model for Requirements Traceability in a Heterogeneous Model-Based Design Process: Application to Automotive Embedded Systems, in: Proceedings of ICECCS, 233–242, 2010.

[38] J. A. B. Montevechi, A. F. de Pinho, F. Leal, F. A. S. Marins, Application of Design of Experiments on the Simulation of a Process in an Automotive Industry, in: Proceedings of WSC, WSC '07, IEEE Press, 1601–1609, 2007.

[39] R. Adler, D. Domis, K. Höfig, S. Kemmann, T. Kuhn, J.-P. Schwinn, M. Trapp, Integration of Component Fault Trees into the UML (2011) 312–327.